A Kernel Machine Framework for Feature Optimization in Multi-frequency Sonar Imagery

J.R. Stack and R. Arrieta
Naval Surface Warfare Center
Panama City, FL 23407, USA

X. Liao and L. Carin
Duke Univ., Dept of ECE
Durham, NC 27708, USA

***Abstract*—The purpose of this research is to optimize the extraction of classification features. This includes the optimal adjustment of parameters used to compute features as well as an objective and quantitative method to assist in choosing *a priori* data collection parameters (*e.g.,* the insonification frequencies of a multi-frequency sonar). To accomplish this, a kernel machine is employed and implemented with the kernel matching pursuits (KMP) algorithm. The KMP algorithm is computationally efficient, allows the use of arbitrary kernel mappings, and facilitates the development of a technique to quantify discriminating power as a function of each feature. A method for feature optimization is then presented and evaluated on simulated and experimental data. The experimental data is derived from low-resolution, multi-frequency sonar and consists of a large feature space relative to the available training data. The proposed method successfully optimizes the feature extraction parameters and identifies the (much smaller) subset of features actually providing the discriminating capability.**

I. Introduction

One of the most substantial challenges in implementing a machine learning algorithm is in deciding what measurements or features to extract from the raw sensor data and present to the learning machine. This process of feature extraction is often done heuristically where the algorithm designer chooses a set of (hopefully informative) features, and the result is often a large feature set relative to the available training data. To appreciate the ease with which this feature space can grow exponentially, consider the low-resolution, multi-frequency sonar application addressed in this paper.

In this application, objects can be insonified at any number of frequency values spanning almost four octaves. The challenge is to determine how many frequencies and what values will produce the most informative feature set. Once the frequencies of insonification are determined, one approach for the creation of features is to make object measurements at each frequency and compute the ratios of these measurements at every frequency pairing. For example, one may measure the width of an object at each insonification frequency and then compute the ratio (or variation in) width at every insonification frequency paring. For this sort of feature, the addition of every new insonification frequency produces an order d^2 growth in feature space for *each* measurement (*e.g.,* object width, depth, *etc*.).

In addition to optimizing sensor settings, many features have parameters that must also be tuned. For example, objects in sonar imagery are often blurred or fuzzy, and the object's true extent may not be obvious. Therefore, an object in an image is often defined by all pixels greater than some threshold below the maximum pixel value. This produces another set of feature extraction parameters that the algorithm designer must choose and somehow optimize. These examples illustrate the challenge faced in designing a feature set and the need for a technique to optimize features during the extraction process.

Traditional feature selection techniques offer methods to choose adequate subsets of features. However, these methods are typically combinatorial searches that involve training the classifier on various subsets of the full feature space and tracking subsequent changes in classifier performance. While this is a useful technique, it is tedious, expensive, and typically only searches a small subset of feature set configurations.

To truly optimize features during the extraction process in a computationally tractable fashion, one would desire the ability to directly measure the discriminating power of each individual feature. With this capability, each feature could be optimized, ranked, and either retained or discarded as non-informative. To this end, this research approaches the feature set design / optimization problem within the framework of kernel learning machines. This framework facilitates the quantification of the discriminating power of each feature and the means to tune its extraction parameters.

II. Kernel Machines

Kernel machines are the product of recent developments in the field of Statistical Learning Theory [1]-[2]. This field is concerned with solving the learning problem, which equates to minimizing risk, $R(\boldsymbol{\alpha})$, as defined in (1). In this equation, L is the loss or discrepancy between the correct answer, y, and the learning machine's estimate of the correct answer, $\hat{y} = f(\mathbf{x},\boldsymbol{\alpha})$ where $\mathbf{x}$ is an input feature vector and $\boldsymbol{\alpha}$ represents the learned parameters. The difficulty in minimizing this risk functional is that a complete enumeration of the joint probability distribution, $p(\mathbf{x}, y)$, is unknown and only a limited training set, $(\mathbf{x}_j, y_j)$ for $j = 1...J$, is available to the learning process.

$$R(\boldsymbol{\alpha}) = \int L\big(y, f(\mathbf{x},\boldsymbol{\alpha})\big)\, dP(\mathbf{x}, y) \tag{1}$$

It is this fundamental issue that is addressed by one of the most significant contributions of this field: the establishment of a bound on the generalization ability of a learning machine. This principle states that with bounded probability the

1-4244-0115-1/06/$20.00 ©2006 IEEE

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
01 SEP 2006	**N/A**	**-**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
A Kernel Machine Framework for Feature Optimization in Multi-frequency Sonar Imagery	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Naval Surface Warfare Center Panama City, FL 23407, USA**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES
See also ADM002006. Proceedings of the MTS/IEEE OCEANS 2006 Boston Conference and Exhibition Held in Boston, Massachusetts on September 15-21, 2006, The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**UU**	**6**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

inequality of (2) holds where the risk functional evaluated on previously unseen data, $R_{Test}(\boldsymbol{\alpha})$, (*i.e.,* generalization ability), is bounded by the risk over the training data, $R_{Train}(\boldsymbol{\alpha})$, plus a function g. In this formulation, g is a function of the number of training exemplars, J, and the capacity of the learning algorithm, h. The capacity of a learning machine measures the complexity or sophistication of the function set from which the learning machine is allowed to employ in modeling the task at hand.

$$R_{Test}(\boldsymbol{\alpha}) \le R_{Train}(\boldsymbol{\alpha}) + g(J,h) \tag{2}$$

While in practice the bound in (2) is difficult to exploit directly, its rather profound implication is that an optimal solution to the learning problem is achieved via an appropriate balance between J and the *capacity* (rather than between J and the number of free parameters in the model). This notion has led to the development of kernel machines, which are expressed in a general form in (3). In this equation, (restricting our discussion henceforth to the classification problem) $\hat{y}_j$ is the estimated class label of the j^{th} data sample, $\boldsymbol{x_j}$, with $y_j \in \{\pm 1\}\ \forall j$.

$$\hat{y}_j = sign\left(\mathbf{w}^T K(\mathbf{x}_j)\right) \tag{3}$$

If this were a linear classifier (*i.e.,* if $\hat{y}_j = sign(\mathbf{w}^T\mathbf{x}_j)$), $\hat{y}_j$ would be formed by simply projecting the data vector onto the weight vector and applying a threshold.[1] In this case, the decision boundary is a linear hyperplane drawn directly in feature space.

Obviously, this approach would hardly be robust as most real-world classification problems require a decision surface more sophisticated than a simple hyperplane. Therefore, a standard approach is to learn a more complex decision surface directly in feature space (*e.g.,* an artificial neural network). However, recalling the desire to quantify and control the capacity from (2), the kernel machine first performs a nonlinear projection, $K(\bullet)$, from feature space into a new space (kernel space) and then learns a simple decision boundary (a hyperplane) in this new space.

The purpose of K is twofold; first, the specification of this mapping places an upper bound on classifier capacity, h. This significantly improves the chances of designing a learning machine with good generalization ability as expressed in (2). Second, an appropriate choice of K should make the problem easier (*i.e.*, a non-separable problem in feature space should be *more* separable in kernel space). This is possible not only because a nonlinear mapping can rearrange the data with respect to the decision boundary but also because K may map to a substantially higher-dimensional space (recall any non-separable problem can be made linearly separable via a sufficient increase in dimensionality [3]).

There are many approaches to solving the general kernel machine form in (3). One of the most popular and well-performing is the support vector machine (SVM) [4]. The general approach of the SVM is to minimize the following functional

$$\frac{1}{2}\|\mathbf{w}\|^2 - \sum_j \lambda_j m_j \tag{4}$$

where λ_j is a Lagrange multiplier for each data point and m_j is the margin. In kernel machines, the margin of a point is defined as the minimal distance from that point to the decision surface, and a common goal is to find $\mathbf{w}$ that maximizes the margin. While the SVM and its variants can produce state-of-the-art results, the optimization of (4) requires solving a quadratic program over the full dimensionality of kernel space using all input data. This renders the SVM computationally intractable for many large real-world problems, and many techniques have emerged to obviate this issue [5]-[6]. Nevertheless, this issue has also prompted many researchers to consider alternate means of solving (3) in a more computationally tractable fashion.

III. Kernel Matching Pursuit

One such alternative is the kernel matching pursuit (KMP) algorithm, which solves (3) with $K(\mathbf{x}_j)$ replaced with $\boldsymbol{\Phi}$, which is the kernel design matrix.[2] In this formulation, each element $\boldsymbol{\Phi}_{ij} = K(\boldsymbol{\theta}_i,\mathbf{x}_j)$ constitutes the (scalar) result of the kernel functional applied to the j^{th} data vector, $\mathbf{x}_j$, and the i_{th} basis function, $\boldsymbol{\theta}_i$. Following this notation, $\boldsymbol{\varphi}(\boldsymbol{\theta}_i.) = \boldsymbol{\varphi}_i$ is the i_{th} row of $\boldsymbol{\Phi}$, which is the kernel functional result of every data vector and the i^{th} basis $\boldsymbol{\theta}_i$.

A. Learning $\boldsymbol{\Phi}$ *and* $\mathbf{w}$

Typically, the full kernel design matrix $\boldsymbol{\Phi}$ is constructed using every data vector $\mathbf{x}_j$ as the set of basis functions $\boldsymbol{\theta}_i$ (*i.e.,* $\boldsymbol{\Phi}_{ij} = K(\mathbf{x}_i,\mathbf{x}_j)\ \forall\ i, j$). Let the full kernel design matrix constructed in this fashion be represented as $\boldsymbol{\Phi}^{Full}$, and note the substantial size of $\boldsymbol{\Phi}^{Full}$ and associated computational issues regarding its manipulation (*e.g.,* traditional SVM algorithms). However, the KMP approach is to iteratively build $\boldsymbol{\Phi}^{(t)}$ by one basis at each iteration, t, starting with an empty matrix at $t = 0$, $\boldsymbol{\Phi}^{(0)} = \{\varnothing\}$. A greedy approach to this construction was first proposed in [7] and several algorithms were developed for its highly efficient computation. However, since the objective herein is feature optimization, a different approach developed in [8], [9] is employed.

In this approach, an error is defined as

$$e = \left[\mathbf{w}^T\boldsymbol{\Phi}^{(t)}\right]^T - \mathbf{Y} \tag{5}$$

Where $\mathbf{Y} = [y_1, y_2 \ldots y_J]^T$, and the objective is to minimize a weighted least squares error

$$E = \mathbf{e}^T\boldsymbol{\Sigma}\mathbf{e} / tr(\boldsymbol{\Sigma}). \tag{6}$$

[1] In practice, the threshold or offset is often learned by augmenting the data vector: $\mathbf{x}_j = [1, x_1, x_2 \ldots x_D]^T$ for a D-dimensional feature space.

[2] In KMP and other newer kernel machines, $\boldsymbol{\Phi}$ is no longer restricted to only Mercer kernels as it is with the SVM.

In this objective function, $\mathbf{\Sigma}$ is a diagonal matrix whose elements place a weight on each data vector and $tr(\bullet)$ is the trace operator. A common choice for the diagonal elements of $\mathbf{\Sigma}$ is the reciprocal of the number of exemplars in that class. This construction of $\mathbf{\Sigma}$ is used to mitigate the effects of unbalanced class sizes in the training set. A well-known minimizer to (6) is the weighted least squares solution

$$\mathbf{w} = \mathbf{M}^{-1}\mathbf{\Phi}^{(t)}\mathbf{\Sigma}\mathbf{Y} \tag{7}$$

where the Fisher Information matrix, $\mathbf{M}$, is

$$\mathbf{M} = \mathbf{\Phi}^{(t)}\mathbf{\Sigma}\left(\mathbf{\Phi}^{(t)}\right)^{T}. \tag{8}$$

While the above equations provide an efficient means to compute the weight vector, the key to feature optimization is in the way $\mathbf{\Phi}^{(t)}$ is constructed. First, the full kernel design matrix $\mathbf{\Phi}^{Full}$ is constructed in typical fashion by using every data vector as the set of bases. Then $\mathbf{\Phi}^{Full}$ is augmented by concatenating a vector of all ones on top of the first row to serve as the offset for the hyperplane in kernel space (*e.g.,* the SVM solves for the offset separately as the parameter b [5]). The empty kernel matrix of iteration zero, $\mathbf{\Phi}^{(0)}$, is populated at iteration one with the offset row from $\mathbf{\Phi}^{Full}$ (*i.e.,* $\mathbf{\Phi}^{(1)} = \boldsymbol{\varphi}_1^{Full}$).

Afterwards, the error for the current iteration, $E^{(t)}$, is computed by solving (5)-(8). The error for the next iteration can then be expressed as

$$E^{(t+1)} = E^{(t)} - \delta\left(\boldsymbol{\varphi}_n\right) \tag{9}$$

where $\boldsymbol{\varphi}_n$ is any basis (*i.e.,* row) from $\mathbf{\Phi}^{Full}$ not yet included in $\mathbf{\Phi}^{(t)}$. To choose the best basis, $\delta(\boldsymbol{\varphi}_n)$ is computed for all remaining bases and the one that maximizes $\delta(\boldsymbol{\varphi}_n)$ is chosen and concatenated to $\mathbf{\Phi}^{(t)}$.

$$\delta\left(\boldsymbol{\varphi}_n\right) = \left[\mathbf{a}^{T}\mathbf{w}^{(t)} - \boldsymbol{\varphi}_n\mathbf{\Sigma}\mathbf{Y}\right]^2 / \left[tr(\mathbf{\Sigma})b\right] \tag{10a}$$

$$\mathbf{a} = \mathbf{\Phi}^{(t)}\mathbf{\Sigma}\left(\boldsymbol{\varphi}_n\right)^{T} \tag{10b}$$

$$b = \boldsymbol{\varphi}_n\mathbf{\Sigma}\left(\boldsymbol{\varphi}_n\right)^{T} - \mathbf{a}^{T}\left(\mathbf{M}^{(t)}\right)^{-1}\mathbf{a} \tag{10c}$$

This process continues until a stopping criterion is met such as reaching threshold on the relative error decrease [9] or the Fisher Information matrix becoming sufficiently rank deficient.

B. Feature Optimization

The function δ in (9) effectively quantifies the decrease in classification error due to the inclusion of the next basis $\delta(\boldsymbol{\varphi}_n)$. The authors of [8] capitalize on this principle by optimizing the mapping into kernel space as a function of discriminating power (*i.e.,* maximize δ as a function of the tunable parameters in K). The work proposed herein takes the similar tact of maximizing δ as a function of the individual feature extraction parameters. Recall each element of $\boldsymbol{\varphi}_i$ is the result of the kernel functional $K(\boldsymbol{\theta}_i,\mathbf{x}_j)$ applied to the j^{th} data vector; also, each element of the j^{th} data vector, x_{dj}, corresponds to the d^{th} feature or measurement of that data exemplar. Therefore, by holding i constant, $\delta(\boldsymbol{\varphi}_i) = \delta(K(\boldsymbol{\theta}_i, \mathbf{x}_j))\ \forall\ j$ is optimized by tuning each of the D measurements that constitute $\mathbf{x}_j$.

This concept forms the basis of the feature optimization technique developed in this paper. Let $\boldsymbol{\eta} = [\eta_1, \eta_2\ldots]$ represent the collection of all adjustable parameters over all D measurements (*e.g.,* frequencies, thresholds, *etc*.). Then for a fixed i, $\delta(\boldsymbol{\varphi}_i)$ becomes a function of $\boldsymbol{\eta}$ and can be optimized. In the following section, this process is illustrated in detail and demonstrated on simulated and experimental data.

However, in practice, not all features are amenable to optimization in this fashion. For example, insonification frequency is a sensor setting, and in many cases, the data acquisition and algorithm training processes are decoupled and occur at different points in time. In this event, a feature set must be developed based on pre-collected training data acquired over a set of pre-determined sensor settings. Here the ability to quantify the relative discriminating power of each individual feature is of great value to the process of feature set design. Through this ability, training data can be collected over a wide range of sensor settings, and the algorithm designer can then efficiently and objectively choose the best sensor configuration(s) within the constraints of the application.

To illustrate this principle, consider the use of a Gaussian functional as a kernel mapping.

$$\Phi_{ij} = K(\boldsymbol{\theta}_i, \mathbf{x}_j) = e^{-\frac{1}{\gamma^2}\left(\boldsymbol{\theta}_i - \mathbf{x}_j\right)^{T}\mathbf{Q}_i\left(\boldsymbol{\theta}_i - \mathbf{x}_j\right)} \tag{11}$$

In this equation, γ is a radius parameter and is a parameter to adjust for optimizing the projection into kernel space. The variable $\mathbf{Q}_i$ is a diagonal matrix whose elements $[q_1, q_2, \ldots q_D] \in [0,1]$ serve as coefficients for each dimension in feature space (*i.e.,* $\mathbf{Q}$ is D by D by I). Therefore, each q_d is adjusted individually for each $\boldsymbol{\theta}_i$, and $\delta(\boldsymbol{\varphi}_n)$ is maximized as a function of $\mathbf{Q}_i$ (as well as $\boldsymbol{\eta}$). In this fashion, the elements of $\mathbf{Q}_i$ reflect the relative discriminating power of each feature.

Table I contains pseudocode for training the KMP algorithm with feature optimization. In practice, the optimization of $\boldsymbol{\eta}$ is typically performed over the first few iterations and the resulting optimum (or mean, *etc.*) value is chosen. The entire algorithm is then repeated with $\boldsymbol{\eta}$ fixed at this value to optimize $\mathbf{Q}$.

IV. Experimental Results

A. Results from Simulated Data

To illustrate this procedure, consider a 2-class problem with each class drawn from 2-dimensional, partially overlapping Normal distributions as illustrated in Fig. 1. From this raw data, a 4-dimensional feature space is constructed as follows:

$\mathbf{x}_1 \sim \mathcal{N}(-1,1)$

$\mathbf{x}_2 = \mathbf{v}_1$

$\mathbf{x}_3 = \mathbf{v}_2 + \mathcal{N}(\text{mean}(\mathbf{v}_2), 3\text{-}\eta_1)$

Table I

PSEUDOCODE FOR TRAINING THE KMP ALGORITHM WITH FEATURE OPTIMIZATION

1. Extract features, $\mathbf{x}_j$, from raw data
2. Compute $\mathbf{\Phi}^{Full}$ (full, *augmented* kernel design matrix)
3. Initialize $\mathbf{\Phi}^{(1)} = \boldsymbol{\varphi}_1^{Full}$
4. Loop
   4.a. Compute $\mathbf{M}$, $\mathbf{w}$, $\mathbf{e}$, & E
   4.b. Check for stopping criteria
   4.c. Choose $\delta(\boldsymbol{\varphi}_n)$ to add to $\mathbf{\Phi}^{(t)}$ as arg max $\delta(\boldsymbol{\varphi}_n)$ $\forall$ n not already in $\mathbf{\Phi}^{(t)}$
   4.d. Optimize $\boldsymbol{\eta}$: maximize $\delta[\boldsymbol{\varphi}_n(\boldsymbol{\eta})]$ for n from Step 4.c
   4.e. Optimize $\mathbf{Q}$: maximize $\delta[\boldsymbol{\varphi}_n(\mathbf{Q})]$ for n and $\boldsymbol{\eta}$ from Steps 4.c-4.d
   4.f. Update $\mathbf{\Phi}^{(t+1)}$ with new $\boldsymbol{\varphi}_n$ computed from optimized $\boldsymbol{\eta}$ and $\mathbf{Q}$

$\mathbf{x}_4 \sim \mathcal{N}(3,1)$.

By design, this feature set has two non-informative dimensions (the 1st and 4th) and a noise term with an adjustable variance added to the 3rd dimension. From the discussion in the previous section, one would expect the optimization of $\mathbf{Q}$ to suggest the only information is contained in the 2nd and 3rd features (*i.e.,* $\mathbf{Q} \approx \text{diag}([0, 1, 1, 0])$) and the optimal value of $\eta_1 \approx 3$.

Table II presents results obtained using 100 randomly drawn samples for training and 100 for testing. The first row in the table presents results for the standard KMP algorithm with no optimization and serves as the baseline. The second row illustrates that the optimum value of $\boldsymbol{\eta}$ significantly improves performance. It can also be seen that the optimized value of $\boldsymbol{\eta}$ (for this finitely sampled training set) is close to its true optimal value. The third row indicates that by optimizing $\mathbf{Q}$ slightly better performance is achieved with fewer basis functions. As discussed in Section II, one of the goals of a kernel machine is to minimize classifier capacity, and minimizing the number of bases directly contributes to this goal.

An extremely valuable piece of information not apparent in the results of Table II is the implication of the values of $\mathbf{Q}$. In the first few iterations of the algorithm, $\mathbf{Q}$ assumed a value that was to be expected (*i.e.,* $\mathbf{Q}_1 \approx [0, 0.5, 0.5, 0]$). However, as the algorithm progressed, the value of $\mathbf{Q}$ began to approach the identity matrix (*i.e.,* $\mathbf{Q}_{11} \approx \mathbf{I}$). This behavior can be explained as follows. As the first few bases are being added to $\mathbf{\Phi}^{(t)}$, the KMP is partitioning the kernel space on a gross scale and clearly should not consider the 1st and 4th non-informative features. However, as the size of $\mathbf{\Phi}^{(t)}$ grows, the KMP is effectively fine tuning the decision boundary, and since there are only a finite number of samples in the training set, it is using every sample in every dimension to best fit the training data.

This insight is extremely valuable to the algorithm designer trying to choose sensor configurations and design a feature

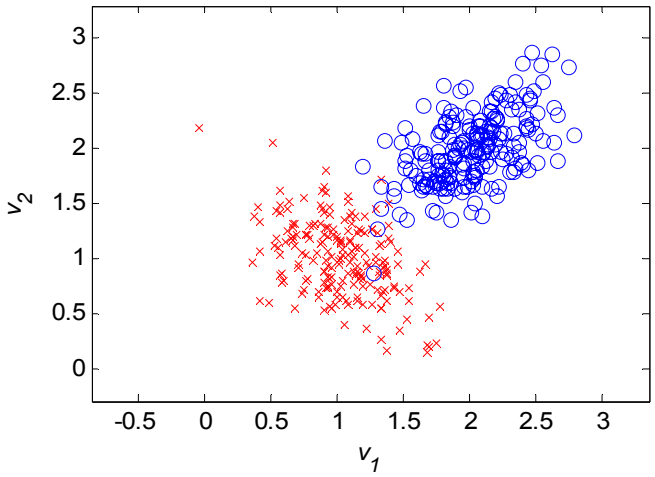

Fig. 1. Simulated data drawn from two normal distributions. Raw data space is 2-dimensional (v_1, v_2).

set. Specifically, the 1st and 4th features *are* helping the KMP better fit the *training* data. However, this is clearly due to finite sample size effects rather than actual information contained in these features. The fact that the 1st and 4th elements of $\mathbf{Q}$ are approximately zero in the beginning of the learning process directly indicates this. Therefore, by observing this behavior in $\mathbf{Q}$, the algorithm designer can replace these features with more informative measurements or eliminate them and reduce the dimensionality of feature space.

B. Results from Multi-frequency Sonar Data

To illustrate this principle on actual data, consider the use of a low-resolution, mechanically steered sonar to classify underwater objects. This is a useful application due to the affordability and availability of these sensors; however, the low image resolution often obviates traditional image-based techniques common in high-resolution imagery. This is illustrated in Fig. 2 where a sphere and cylinder are insonified, and it is not obvious how to correctly classify the objects based on the imagery alone. It has been demonstrated that a useful approach to this problem is to insonify the objects at multiple frequencies and examine the backscatter variations over frequency [10], which is the approach adopted herein.

The target set for this experiment consists of a cylinder and sphere of approximately the same diameter and a cylinder length of approximately 3X the diameter. The targets are insonified multiple times at 4 frequencies, $\{f_1, f_2 \cong 1.5f_1, f_3 \cong 2f_1, f_4 \cong 3f_1\}$, and the cylinder over 3 aspects, {0°, 45°, 90°}. For each object in each image, 17 measurements are made and used to construct the features. Of these 17 measurements, 5 are textural based and are estimated from the gray level co-occurrence matrix (GLCM) [11]. The GLCMs are computed in standard fashion using multiple offsets in the vertical direction; the GLCM statistics are then estimated for each offset and averaged. The remaining 12 measurements are geometric, and all measurements are listed in Table III. The

Table II

EXPERIMENTAL RESULTS FROM SIMULATED DATA

	Number of Misclassified Exemplars (%)		Number of Basis Functions Used
	Training	Testing	
No optimization $\boldsymbol{\eta} = [0]$	9 (4.5%)	20 (10)	16 (8)
Optimize $\boldsymbol{\eta}$ only $\boldsymbol{\eta}_{opt} = [2.82]$	3 (1.5)	10 (5)	16 (8)
Optimize $\mathbf{Q}$ with $\boldsymbol{\eta} = [2.82]$	1 (0.5)	10 (5)	11 (5)

bottom 4 geometric measurements in the right-hand column are derived from an estimated ellipse that best encloses the object while $A^2(Z)$ is a corrected Anderson-Darling statistic that measures the quality of Gaussian fit [12].

After the measurements in Table III are made, the features are computed by taking the ratio of each measurement for each of the 6 possible frequency pairings—for $\mathbf{F} = \{f_1, f_2, f_3, f_4\}$, the pairings are $\{f_1/f_2, f_1/f_3, f_1/f_4, f_2/f_3, f_2/f_4, f_3/f_4\}$. These ratios form the actual features presented to the classifier. To train and test the classifier, a set of 70 exemplars is randomly divided into 39 training and 31 testing exemplars (where one exemplar is a set of images over $\mathbf{F}$). This random selection process is repeated 3 times to form 3 different data sets.

As with the simulated data, the first step in the training process is to optimize $\boldsymbol{\eta}$. For this feature set, $\boldsymbol{\eta} = \eta_1$ is a threshold (in dB) used in extracting most of the geometric measurements. Specifically, to make measurements on a object in an image, the extent of the object must be specified. This is accomplished by considering the object to be comprised of all pixels η_1 dB below the maximum pixel value in the object. The optimization of η_1 is performed and the average value over the 3 data sets is found to be $\eta_1 = 3.9$ dB.

The next step is to determine which features are actually providing discriminating power and which ones can be discarded. The 17 object measurements computed over the 6 frequency pairings result in a 102-dimensional feature space. While this is clearly too large of a space to characterize with only 39 training exemplars, this imbalance between training samples and feature space dimensionality is common. As is often the case, the measurements of Table III represent an intuitively compiled list of hopefully informative measurements and the frequency pairings represent a subjective, *a priori* estimate limited by practical data collection constraints. However, it is unclear during the initial specification of these features which ones will actually prove useful in classification.

The process of identifying the useful features begins by training the classifier (using the optimal η_1) on the full (102 dimensional) feature set. This serves as a baseline for comparison and the results are listed in the first group of Table IV. The next step is to optimize $\mathbf{Q}$ and use it to prune

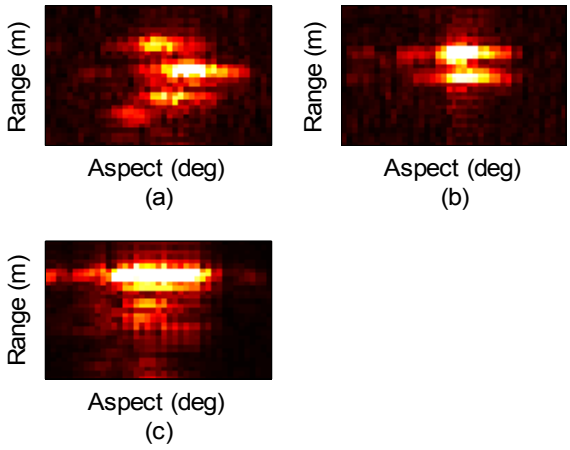

Fig. 2. A sphere (b) and hollow cylinder ((a) = Aspect 2 & (c) = Aspect 3) are insonified at the same frequency.

the feature set. Unfortunately, with such a large feature space, $\mathbf{Q}_i = \mathbf{I}\ \forall\ i$ and this optimization step is not helpful. This phenomenon can be understood by recalling that given a high enough dimensionality any two classes can be perfectly separated. In such an overly large feature space, the classifier is using all dimensions equally to fit (over train) the training data. Therefore, the initial reduction in feature space is done as follows.

The classifier is trained 4 separate times by redefining $\mathbf{F}$ as only containing 3 of the 4 original frequencies. It was found that $\mathbf{F} = \{f_1, f_3, f_4\}$ reduced the feature space to 51 dimensions and caused no decrease in classifier performance. This is illustrated as the second group of results in Table IV. For comparison, the third group of results illustrates what happens if only two frequencies are used. In this case, performance improves for some of the data sets with a change in the number of basis functions used.

Afterwards, $\mathbf{F}$ is fixed as a subset of the original frequencies, $\mathbf{F} = \{f_1, f_3, f_4\}$, and $\mathbf{Q}$ is optimized. With the dimensionality of feature space more commensurate with the size of the training set, the optimization of $\mathbf{Q}$ proceeds as expected and $\mathbf{Q}_{opt}$ is inspected to identify and remove the non-informative features. This procedure results in the reduced feature set referred to in the fourth group of Table IV. The membership of this reduced feature set is indicated in Table III where only the measurements followed by a ♦ are included and some of these are only computed for a subset of frequency ratios (indicated by a set of ratios following the ♦). From these results, significant gains in performance are achieved using almost half of the original number of basis functions.

As a final check, the previously omitted frequency pairings involving f_2 are replaced in the reduced feature set and found to add no value to classifier performance. Therefore, it is concluded that the omission of f_2 was a valid choice. In the last step, the projection into kernel space is optimized to produce the best performance with the fewest number of basis

Table III

OBJECT MEASUREMENTS USED TO CONSTRUCT FEATURES

Textural Measurements	
GLCM Contrast	GLCM Homogeneity
GLCM Correlation	GLCM Entropy
GLCM Energy ♦	
Geometric Measurements	
Number of Object Peaks ♦ $\{f_1/f_3\}$	Width (deg)
Range Location of Largest Object Peak ♦ $\{f_1/f_3,\ f_3/f_4\}$	Depth (m)
Peak Pixel Value / Average Background Value ♦ $\{f_1/f_4\}$	Eccentricity
Kurtosis ♦	Euler Number ♦
Skewness ♦	Solidity ♦ $\{f_1/f_3\}$
$A^2(Z)$ ♦	Orientation ♦ $\{f_1/f_3,\ f_3/f_4\}$

functions. This is achieved by optimizing γ in the same fashion as η_1, and this result is in the last group of Table IV. While it may be noticed that γ in (11) could be absorbed into $\mathbf{Q}$ and the two optimized together, this is not done for the following reason. The optimization surface of $\delta[\varphi_n(\mathbf{Q})]$ is complex, nonlinear, and highly dependent upon initial conditions while $\delta[\varphi_n(\gamma)]$ is less complicated and much less dependent on perturbations in initial conditions. Therefore, the two are kept separate, $\mathbf{Q}$ is optimized first with an initial condition of $\mathbf{Q} = \mathbf{I}$, and γ is then optimized with an arbitrary initial condition.

V. CONCLUSIONS

In conclusion, this work has presented a method for optimizing the feature extraction process. This method is amenable both to extracting parameters that can be adjusted during the learning process as well as objectively determining data collection parameters (*e.g.,* sensor settings) *a priori*. This provides the algorithm designer a powerful tool for designing feature extraction algorithms, specifying optimal sensor settings, and performing feature selection based on a quantitative measure.

ACKNOWLEDGEMENT

This work was partially supported by the Very Shallow Water / Surf Zone MCM Reconnaissance project sponsored by Dr. T. Swean of the Office of Naval Research, ONR 321OE.

Table IV

EXPERIMENTAL RESULTS FROM SONAR DATA

Data Set	Feature Set	Number of Misclassified Exemplars (%)		Number of Basis Functions Used
		Training	Testing	
1	Full Feature Set $\mathbf{F} = \{f_1, f_2, f_3, f_4\}$	0 (0%)	8 (26)	12 (30)
2		0 (0)	8 (26)	12 (30)
3		0 (0)	8 (26)	12 (30)
1	Full Feature Set $\mathbf{F} = \{f_1, f_3, f_4\}$	0 (0)	8 (26)	12 (30)
2		0 (0)	8 (26)	12 (30)
3		0 (0)	8 (26)	12 (30)
1	Full Feature Set $\mathbf{F} = \{f_1, f_4\}$	0 (0)	1 (3)	16 (40)
2		0 (0)	8 (26)	10 (25)
3		0 (0)	7 (23)	10 (25)
1	Reduced Feature Set $\mathbf{F} = \{f_1, f_3, f_4\}$	0 (0)	1 (3)	6 (15)
2		0 (0)	8 (26)	7 (18)
3		0 (0)	4 (13)	7 (18)
1	Optimized $\gamma = 1.16$	0 (0)	3 (10)	5 (13)
2		0 (0)	4 (13)	3 (8)
3		0 (0)	6 (19)	4 (10)

REFERENCES

[1] V.N. Vapnik, "An overview of statistical learning theory," *IEEE Trans. Neural Networks*, vol. 10, no. 5, pp. 988-999, Sept. 1999.

[2] V.N. Vapnik, *The Nature of Statistical Learning Theory*, New York: Springer, 2000.

[3] R.O. Duda and P.E. Hart, *Pattern classification and scene analysis*, New York: Wiley & Sons, 1973, pp. 69-70, 134-138.

[4] C. Corets and V.N. Vapnik, "Support vector networks," *Machine Learning*, vol. 20, pp. 273-297, 1995.

[5] K-R Muller, S. Mika, G. Ratsch, K. Tsuda, and B. Scholkopf, "An introduction to kernel-based learning algorithms," *IEEE Trans. Neural Networks*, vol. 12, no. 2, Mar. 2001.

[6] B. Scholkopf and A.J. Smola, *Learning with Kernels: Support vector machines, regularization, optimization, and beyond*, Cambridge, Mass.: MIT Press, 2002, pp. 279-329.

[7] P. Vincent and Y. Bengio, "Kernel matching pursuit," *Machine Learning*, vol. 48, pp. 165-187, 2002.

[8] X. Liao, H. Li, and B. Krishnapuram, "An M-ary KMP classifier for multi-aspect target classification," *Proc. IEEE ICASSP*, vol. 2, pp. 61-64, May 17-21, Montreal, Canada, 2004.

[9] X. Liao and L. Carin, "Radial basis function network for multi-task learning," *Neural Information Processing Systems*, http://www.nips.cc, Dec. 5-7, Vancouver, Canada, 2005.

[10] J.R. Stack, G. Dobeck, and C. Bernstein, "Multi-frequency backscatter variations for classification of mine-like targets from low-resolution sonar data," *Proc. IEEE/MTS OCEANS*, vol. 1, pp. 218-222, Washington DC, Sept. 19-23, 2005.

[11] A. Baraldi and F. Parmiggiani, "An investigation of the textural characteristics associated with gray level co-occurrence matrix statistical parameters," *IEEE Trans. Geoscience & Remote Sensing*, vol. 33, no. 2, pp. 293-304, Mar., 1995.

[12] M.A. Stephens, "EDT statistics for goodness of fit and some comparisons," *American Statistical Assoc.,* vol. 69, no. 347, pp. 730-737, Sept., 1974.